

Préparation séance 3

PARTIE 1 :

Question 1 :

```
< ?php
$time_start = microtime(true) ;
(Séquence d'instruction)
$time_stop = microtime(true) ;
$time = $time_stop - $time_start ;
?>
```

Question 2 :

Un index dans une base de données permet de stocker les données de chaque colonne de manière ordonnée afin d'y accéder efficacement. Les requêtes sont par conséquent plus rapide à être exécutées lors de l'ajout d'un index dans une colonne.

Le fonctionnement de cet index est simple. Prenons l'exemple de la table suivante. Nous ajoutons une recette de cuisine à chaque fois que nous en avons testé une :

Numéro	Nom	Difficulté	Temps
1	Poulet au curry	3	55min
2	Lentilles fumées	1	20min
3	Poire flambées	1	10min

Pour chaque colonne il est possible de créer un index. Si nous décidons de créer un index sur la colonne nom nous aurons la colonne indexée suivante :

2
3
1

Le premier nom par ordre alphabétique correspond à Lentilles fumées soit le numéro 2 suivie de Poire flambées qui est le numéro 3 et ainsi de suite.

PARTIE 2 :

Question 1 :

Le log de requête et un tableau contenant toutes les requêtes qui ont été exécuté pour la requête en cour. Ce tableau est accessible via la commande `getQueryLog()` et peut être désactivé par la commande `disableQueryLog()`.

Question 2 :

Le problème des N+1 query est un problème qui se pose dans une base lorsque 2 tables ont une relation entre elles de la formes BelongsTo. Prenons comme exemple 2 tables, auteur et livre, sachant qu'un livre a été écrit par un auteur. Lorsque l'on veut récupérer les auteurs des 50 premiers livres, on va d'abord récupérer le 1^{er} livre puis ensuite on va parcourir la liste des 50 premiers livres depuis le début (donc en incluant le livre "originel") pour récupérer les auteurs.

Ce problème peut être résolu grâce à la commande "with" et l'argument auteur (qui est le nom de la colonne appartenant à la relation BelongsTo) qui va simplifier les requêtes qui seront exécutées à deux.